



TITLE:

<講演3>スーパーコンピューターが 拓く未来

AUTHOR(S):

中島, 浩

CITATION:

中島, 浩. <講演3>スーパーコンピューターが拓く未来. 京都大学附置研究所・センターシンポジウム: 京都からの提言-21世紀の日本を考える (第7回) 「明るい未来像」 2013, 7: 29-46

ISSUE DATE:

2013-02-26

URL:

<http://hdl.handle.net/2433/172942>

RIGHT:

講演3

スーパーコンピュータが拓く未来

学術情報メディアセンター教授 中島 浩



ご紹介いただきました中島でございます。きょうは、「スーパーコンピュータが拓く未来」ということでお話しさせていただきます。

スーパーコンピュータは、我々、スパコン、スパコンと言っていて、2、3年前に「スパコンで世界一とる必要があるんですか」みたいな、何かそういう話がありましたんで、スパコンという言葉は聞かれたことがあるかと思いますが、それに関しているいろんな話をさせていただくということです。

ここは、とにかく神戸ですので、しかも、ポートアイランドで、ポートライナーに2つ乗ると、「京コンピュータ前」という駅があるんです。日本でコンピュータの名前がついた駅というのは、ここしかないんです。というぐらい、京というのは、最近有名になっているんですが。

その京というのは世界で一番速いスーパーコンピュータなんですが、何でそれがスーパーなんやというお話です。言っておきますけど、「けい」というのは京都の京という字を書きますが、「きょう」とは呼ばなくて、「けい」と呼びます。京都とは関係がありません。神戸です。

それがどのぐらいスーパーなんやという話と、じゃ、それ自分で、つくろうと思ったら、どうやってつくるんですかという話、その辺をまとめて、結局スパコンというのは、どういうものでしょうかと、そんなに難しいものではありません。

じゃ、スパコンを使って未来は拓くのかと、特に明るい未来は拓けるのかという話をさせていただくんですが、そこで計算科学という言葉が出てきます。これはコンピュータを



スーパーコンピュータが拓く未来

京都大学
学術情報メディアセンター
中島 浩

© 2012 H. Nakajima

(図1)



目次

© 2012 H. Nakajima

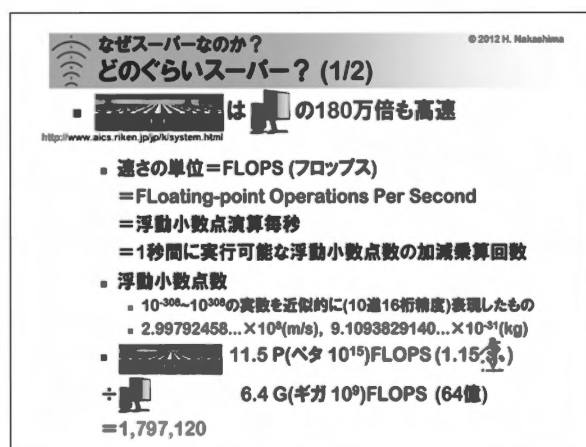
- (京)はなぜスーパーなのか？
 - ・ どのぐらいスーパー？
 - ・ スーパーにする方法
 - ・ 結局スパコンとは？
- (明るい)未来は拓くか？
 - ・ 計算科学＝第3の科学
 - ・ スパコン(が得意な計算)で拓く未来
 - ・ スパコン自身の未来は？

(図2)

使った科学のことなんです、これは第3の科学と呼ばれていまして、これに関してちょっとお話しします。

じゃ、スパコン使って計算をするんですけども、どんな未来が拓けるのかなという実例をほんの少しだけお話しします。もう一つ重要なポイントとして、スパコン自身、未来はあるのかという、これちょっと結構クリティカルな話なんです、させていただきます。

まずスパコンは、どれだけ速いんですかという話がありまして、スパコン、これが京の全体像（図3）。冷蔵庫みたいなのが山のように並んでいるんですが、これは、パソコンの180万倍ぐらい速いということになっています。パソコンもいろいろありまして、ピンからキリまでありますので、いろいろあるんですが、この場合のこのパソコンは、一番安い4万4000円ぐらいだったかなという一番安いパソコンですが、その180万倍ぐらい高速であると。



(図3)

速い、速いといって、別にスパコンは、そこら中を動き回ったりしませんし、飛んだりしませんから、何で速さを測るんですかという話があるんですが、その単位としてフロップスという非常に読みにくい単位を使います。これは、英語の Floating-point Operations Per Second というものの FLOPS をとったものです。これじゃ、全然何のことか分かりませんので、日本語に訳すと、「浮動小数点演算毎秒」ということになって、余計分からなくなる。

もう少しかみ砕くと、1秒間に実行可能な浮動小数点数の加減乗算回数と、どんどん分からなくなるんですが、1秒間は分かりますよね。回数も分かりますね。加減乗算、いいですね。足し算、引き算、掛け算。で、おい、割り算どないしたんやといって、割り算は難しいんですが、もちろんできますけども、できますけども、速さを比べるときは、速さを測るときは足し算か、引き算か、掛け算です。どれでも同じだけできるんですが、この回数です。実行可能な演算というのは、1秒間にできる演算の数です。

この浮動小数点数って何やねんという話になりまして、ちょっとだけせつかく来たんでご説明しておきますと、コンピュータというのは基本的に整数というのしか扱えません。いいですか、整数はご存じですね。1、2、3、4、5、6、0、-1、-2、-3、-4。しかも無限は扱えませんので、かなり大きな数ですけども、どこか適当な大きさまでしか扱えません。

ところが、コンピュータで計算するときは、物理現象、例えば、温度とか、圧力だとか、速度だとか、重さだとか、そういうのを扱うわけ。これは、高校生に返って思い出し

ていただくと、実数というやつです。 $\sqrt{3}$ とか、 π とか、実数なわけなんですけど、実数を扱わなければいけない。実数は、皆さんご記憶のように、いくら小数点の下に数を並べていっても終わらない、絶対終わらない。実数というのは、ちょっと言いすぎですね。無理数というのがあって、 $\sqrt{3}$ とかは終わりませんね。

そんな数、どうやって整数しか扱えんやつが扱うんやというのと、これを近似的にやるしかない。実数を近似的に表現したものです。どのぐらいの大きさの実数が表現できるかというと、10のマイナス308乗から10の308乗という、これは、めちゃくちゃにとんでもない数です。ほんとにめちゃくちゃにとんでもないんです。

どのぐらいにとんでもないかというと、今この宇宙にある素粒子とって、一番小さいやつがあるんですが、それが何やというのは置いておいて、この数が大体10の85乗ぐらいやろうということです。その85の4倍ではないですから、308乗というのは。その素粒子1個1個の中に宇宙の素粒子がまたみんな入っていて、その中にまた宇宙の素粒子が、みんな入っていると、10の300乗ぐらいになるということですから、とんでもなく大きな数から小さい数まで計算できるというわけです。

精度、10進16桁精度、10進数で16桁の精度があるというわけです。ちょっと例を示していますが、これは光の速さで2.99いくら掛ける10の8乗。こういうふうに、これが何桁あるかというのが精度で、あと適当に大きな数を掛けてやればいい。

小さい数は、これ電子の質量ですが、9.1何とかかんとか掛ける10のマイナス31乗と、こういうふうに、こいつとこいつと分けて表現するんですが、これ見たら16桁もないですね。パッと見て、これ10桁ぐらい、これ12桁ぐらいかと思いますが、これ以上は実は分かっていない。光の速度だって、電子の質量だって、細かく測れば分かるんだと思いますけども、そのぐらい人間が頑張っても、光の速度が、このぐらいの精度しか求められないので、16桁もあれば何とかなるやろうというような格好で、近似的といっても、かなり本物に近い感じで表現します。で、ようやくこの浮動小数点数が分かったというわけです。

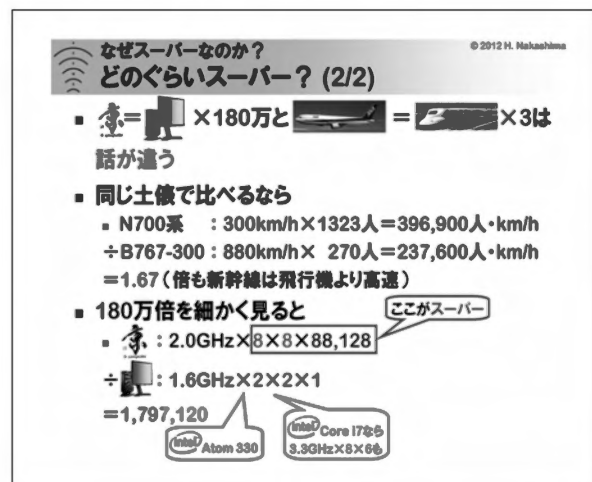
それが京の場合は11.5ペタフロップス。ペタって何やという話があるんですが、ペタというのは、これはキロとか、メガというのが次に来まして、で、ギガ、ギガヘルツとかギガバイトとか、ギガビットとか聞かれたことがあると思いますけども、ギガ。その次、テラ、例えば、ハードディスクレコーダー2テラバイト、3チャンネル何たらかんたら、それがテラです。

その次に来るのが、ペタです。3桁ずつ上がって行ってペタ。それで10の15乗フロップスです。10の15乗掛ける11.5というのが、1.15京。

京というのは兆の1万倍です。その次に垓（がい）が来て、秭（し）が来て、穰（じょう）が来てとか、何か無量大数とかあるわけなんですけど、ということなので、これは京という名前のコンピュータ、大体10ペタフロップス、1京フロップスできるようになる。

パソコンは6.4ギガフロップスぐらい、この一番安いやつです。これが64億ですので、こいつをこいつで割ってやると、179万いくらということで、約180万と。これをご覧になった方は、きっとパソコンってノタノタ、ノタノタ計算しておるんやろうと、京はピューッと行くんやろうと思われると思うんですが、それは違います。(図4)

京がパソコンの180万倍速いというのと、飛行機が新幹線の3倍ぐらい速いというのとは話が違います。同じ土俵で比べるとすると、N700系の新幹線というのは時速300キロでブーンと走るわけですが、これに1323人座れます。立たせると倍ぐらい乗れると思いますが、座るとこれだけです。そうすると、これを掛け算すると、39万6900人キロメートル/hとか、何かよく分からない数になってきます。



(図4)

じゃ、ボーイング、全日空の767ですが、これは時速880キロメートル、これは3倍ぐらい速いんですけども、当然乗る人数が少なくて、270人しか乗れません。飛行機は立って乗ってはいけませんから、もう絶対270人しか乗れない。これを掛け算してやると、23万7600人キロメートル/hということになりまして、新幹線は飛行機よりも1.67倍速い。ええっという不思議な答えになります。これが重要なんです。

じゃ、京が180万倍速いというのはどうなっているかというと、このキロメートル/hというあたりがギガヘルツというところに効いてきまして、京は2.0ギガヘルツという周波数で動きます。これが動作の単位です。この1つの動作の単位に、8×8×8万8128とかいう、すごい値がかかって、これを計算すると、先ほどの11.5ペタフロップスとかになるわけです。

パソコンの場合は、安いパソコンなら1.6ギガヘルツとかいうので動いていまして、このかかっている数がえらいかわいい、2×2×1と。なので、ここの違いで来ている。この1.6ギガというのは一番安いやつですから、ちょっと高いのを買おうと、この2.0ギガが、あっという間に超えて、3ギガとか、4ギガまではちょっと聞きませんが、そんなものも買えます。ですので、ここを実は京というのは、普通の安いパソコンよりもちょっと偉いけども、ちょっと高いパソコンには負けると、その代わり、ここにガガガガッとすごい数がかかっているのが、ポイントです。

ですから、さっきの新幹線がたくさん人を運べるというのと同じように、何か一つの単位でたくさん何かができる。一度にたくさんできるというのがみそなわけです。

もう一遍こんなことを書いていますが、これ、どのぐらいすごいことなのかということですが、2ギガヘルツというのは、0.5ナノセカンドに1回何か波が起こるんですね。0.5ナノセカンドって何やといわれると、また、20億分の1秒ぐらいなんですけども、光が大体15cmぐらい飛びます。0.5ナノセカンドで、光がピュッと飛ぶわけです。(図5)



なぜスーパーなのか？
スーパーにする方法

© 2012 H. Nakashima

- $f = 2.0\text{GHz} \times 8 \times 8 \times 88,128 = 11.5\text{PFLOPS}$
→ 0.5nsの間に $8 \times 8 \times 88,128$ 個の演算を同時に行う
- スーパーにする方法
 - 周波数(2.0GHz)↑
... 最初のスパコンは 80MHz (<携帯電話)だった
 - 並列演算数(×8)↑ ... SIMD, Super Scalar (SS)
... 同じ種類の演算を違うデータに対して同時に行う
 - プロセッサ内並列計算(×8)↑ ... マルチコア
... 独立して計算可能なプログラム中の塊を同時実行
 - プロセッサ数(×88,128)↑ ... MPP
... より独立性が高いプログラム中の塊を同時実行

(図5)

その時間の中で一つ演算するだけでもすごいんですが、京は0.5ナノセカンドの間に、これだけの数の計算を同時にやります。同時にやることができます。頑張れば。この数が大体500万ぐらいになるんですけども、要するに、500万個いっぺんにガツと計算できるというところがすごいわけです。

まず、この4つの数を分解していくんですが、これ、どれを増やしても、この何とかペタフロップスがガンと上がっていくわけです。じゃ、一番単純には、この周波数を上げるということをやっていけばいいわけなんですけど、確かに上がってきた。最初のスパコンというのは、35年ぐらい前にできたんですけども、このときは80メガヘルツというかわいらしい周波数で、これ2.0ギガに対して、25分の1ぐらいです。

ちなみに携帯電話、こんな遅いスピードで携帯電話に動かれると、携帯電話の中、当然コンピュータが入っていますから、そんな携帯電話は使い物になりません。もう全然使えません。携帯電話は、この10倍まで行かないと思いますけど、数倍のスピードで動きます。だから、ああ、そうか、これが上がったんやなという話が1つです。

それから、この並列演算数(×8)というのは、SIMD(シムディー)とか、スーパースカラとか、この辺のわけの分からん言葉は、帰って、グーグルをひっぱたくと言うことでやっていただくとして、これはかみ砕いて言うと、同じ種類の演算を違うデータに対して同時にやる。足し算を同時に4つやる。同じ数を足しておってもしょうがないですから、答え、みんな一緒ですから、いろんな違った数の足し算を同時に4つやるとか、8つやる。これは8つですね。8つやります。これで掛ける8ができます。

次は、「プロセッサ内並列計算」と書いていますが、マルチコアとか呼んでいるんですが、パソコン買ってきて、壊れることを覚悟して蓋を開けていただくと、何かこれぐらいの四角いものが入っています。それを壊すと、パソコンは絶対動きません。それがプロセッサといいます。「インテル入ってる」とか、何かそういうやつです。その蓋は絶対開きませんが、そのプロセッサのチップの蓋を、またガツと開けてやると、中に本当のチップが隠れています。2センチ角とか、そのぐらいの。

その中にコア、CPUコアとかいうんですけども、いわゆるコンピュータ、小さいコン

コンピュータが京の場合8つ入っている。この8つの小さいコンピュータが、共同作業で何かプログラムというものを実行するわけなんですが、プログラムの中には順番にやらなければいけないというやつと、同時にやってもいいやつみたいなのがありまして、同時にやってもいいやつは同時にやる。やっちゃいけないこともあります。

最後の手段というのがありまして、このプロセッサ、パソコンの中に1個とか入っているやつを取りあえず8万ぐらい並べれば、たくさんになるわけです。この、ここがすごいわけなんですが、これMPP、Massively Parallel Processor というんですけども、このマルチコアと同じような話なんですけど、もっと独立性の高いプログラムの固まりを同時に実行する。

言葉で聞いても、よう分からんよねという話がありますので、ちょっとよい例かどうかは分かりませんが、スキー場へ行くとリフトがありますね。昔、行ったという感じでいいんですけども、山がありまして、リフトがある。このリフト、たくさん人が並んでいるので、これを上にパッと上げる、戻ってくるというのがありますけども、上に上げる輸送能力、これを上げたい。(図6)

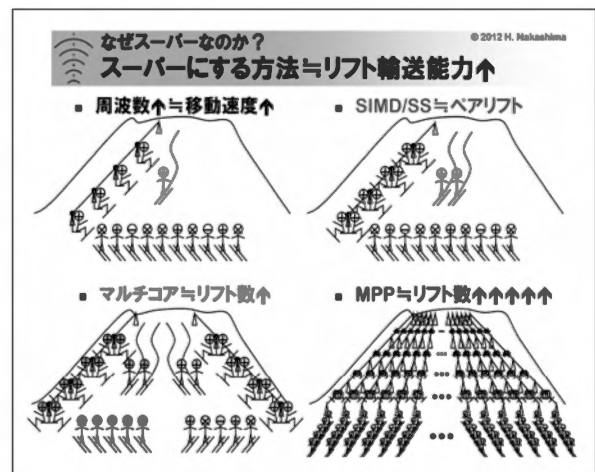
どうやったら速く上がるかと、誰でも考えるのは、ビューンと、このスピードを上げる。けど、リフトですから、これ時速500キロのリフトとかつくと、もうそこら中で死体が転がりますので、それは無理とかいう話もあるんですが、取りあえず一つの方法としては、これを上げる。

輸送能力ですから、ある特定の人がここへ到達する時間を短くするというのは、これしかない。これを速くするしかないんですが、別の方法として、ここの山の上に、どんどん、どんどん人が増えていく速度というのを稼ぐには、ペアリフトにすればいい、倍速くなる。同じスピードで上がっていても、山の上で見ていると2人ずつ下りてくるので、何か速くなったような気がする。これがさっきのSIMDとかの並列演算です。

マルチコアみたいなやつというのは、じゃ、どうするのか、もっと速くしたい。じゃ、リフトを2つつくればいいのか、簡単な話ですね。これがマルチコア。で、最後にここまで来るとちょっと無理があるんですが、もう全山リフトでガーッと埋め尽くす。どこを滑ったらええんか分からんぐらい全山埋め尽くしたのが京みたいな、こんな感じですね。

だから、この辺までは、実はパソコンでもやっていて、京ならではのというのは、全山リフト埋め尽くしというところなんです。

で、まずは周波数を上げればいいのかんという話があって、実際上がってきたんです。



(図6)

これ1975年から最近までの、これインテルという会社の名前はご存じ、「インテル入ってる」ですが、その主力プロセッサの周波数。最初は5メガヘルツぐらいから始まってきまして、どんどこ、どんどこ、どんどこ、どんどこ、ガーッと伸びてきた。(図7)

これ、よくこのトークに出てくるんですが、横軸は5年5年5年・・・とリニアスケールというやつですね。縦軸は10倍

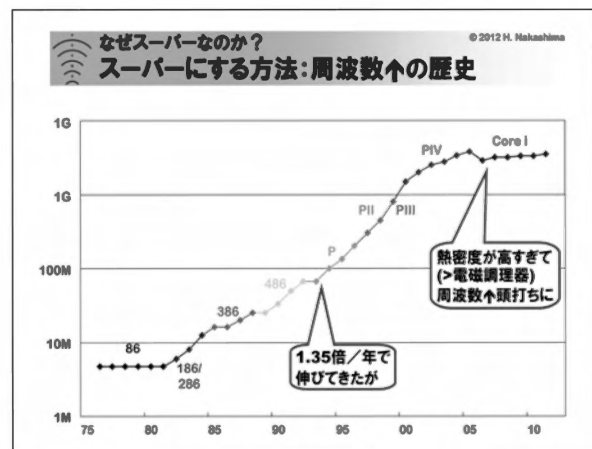
10倍10倍・・・となっていますので、このグラフの上で何となく直線上に上がっていくというのは、倍々ゲームで上がっていく。これを指数関数的に上がるとかいうわけなんです。というんでもない上がり方をしてきたわけです。それがガーッと上がってきて、このままゴーッと行くかなと思っていたわけなんです、これが行かない。

1980年代から90年代にかけて、年率1.35倍というスーパーインフレで上がってきました。毎年35%物価が上がるみたいな話です。だから、ものすごい勢いで上がってきたんですが、これがこの辺でホットプレートになって、もう最近では電磁調理器級の熱密度ということになっています。ホットプレートも熱いけど、電磁調理器はもっと熱い。何でかという、やかんに水を入れてホットプレートの上にポーンと置いていても、絶対お湯は沸きません。ホットプレートはお肉しか焼けない、芋でも焼けますけども、お湯は沸かせない。電磁調理器は沸かせるんですね。

お湯を沸かすためには、かなりの熱密度でたたいてやらなければいけない。これがもう電磁調理器を超えているんです。1990年代の半ばで確か超えたと思うんですけども。これが放っておくと、このままパーッと行くと、あんまり原子炉の話はしたくないんですが、原子炉の何か一番熱いところとかですね。30年たつと、太陽の表面になってとか、何かそんなことになって、そんなもんつくれるはずないわけです。もう溶けちゃうわけです。

これでちょっと、どうしようもないということになっていまして、この10年ぐらい、この周波数は上がらないということになっている。ですから、周波数上げればいいじゃんというのは、何十年前、10年前ぐらいまではそれでやってきたんだけど、もうそれは使えないということになっています。ですから、スーパーコンピュータをつくるには、ほかの手を考えるしかない。

じゃ、さっきの並列何たらといっていたやつ、どうやってやるんやと、そんな都合よく同時にやれる計算なんてあるんかいという話になるんですが、ウー、こんなもん見せられて嫌でしょうが、これ中学生レベルです。中学校か、高校1年ぐらいの算数ですから、頑張ってみましょう。3元連立一次方程式。2元は分かるけど、3元どうやったっけ、同



(図7)

じですから、解き方は一緒なので、何か x とか、 y とか、 z 、好きなやつ消せばいいんですね。で、 x を消す。 x を消すためには、この x の係数が同じになるように、片々何かを掛けて、それで何か片々引くみたいなことをやると消えるんです。(図8)

人間だったら、これ最小公倍数になるように、何か6掛けて、4掛けて、3掛けてみたいなことをするんですが、コンピュータは間抜けですから、これで割ったら、み

んな一緒になるやんけみたいなことで、この式を2で割る、この式を3で割る、この式を4で割るということをまずやります。そうすると、 x の係数がみんな同じになるというわけですね。

で、この2で割る、3で割る、4で割ると、3つ3つ3つ、全部で9つ割ることになるんですけども、これは、どれからやってもいいんです。別にこの順番でやってもいいし、これやって、これやって、これやって、これをと、どの順番でやってもいい、ということは同時にやってもかまわない。あと、割り算下手、割り算大変とっていましたが、よく考えると、これ2で割るんじゃなくて、2分の1を掛けるとしてもいいので、掛け算得意ですから、これ全然平気。

今、係数がそろいましたと、そうすると、この式からこの式を引いて、この式からこの式を引くと、1個減るはずであると、これがここに書いてある。ちょっとぐちゃぐちゃした式になっていますが、書いてある内容は中学生の内容です。

この中にも、これとこれを足す、これからこれを引く、これからこれを引く、これとこれ足す、これ引く、これ引く、これというのは、全部どれからやっても構いません。これ、どれからやってもいい加減算というのがあるわけです。これをやってやると、式が2つになりましたので、もう1回今度は y を消しに行くと z が求まって、で、 z が決まると y が決まって、 y が決まると x が決まるというみたいなことで、方程式が解けたわけです。 $z = 3$ 、 $y = 2$ 、 $x = 1$ と出てくる。

こういう3元連立一次方程式を解く中でも、こういうふうに同時にできることというのがたくさんあるということが分かると思います。コンピュータも実際同じで、多少トリックはありますが、基本的に同じような考え方で解いています。

次に、同時演算じゃなくて、並列計算みたいな話なんですが、さっき言いましたように、これは京の中のチップの写真で、ここに何か丸くかこって書いてありますが、これがCPUコアといわれる小さいコンピュータです。(図9)

[illegible]

(圖 8)

このままでは見にくいので、この四角いやつをCPUコアだと思えば、8つ並んでいまして、メモリというものがついています。京の場合、16ギガバイトのメモリというのがついていて、これは共有メモリという名前がついている。これは何で共有かというと、この小さいコンピュータも、この小さいコンピュータも、このメモリを読んだり書いたりすることができる。

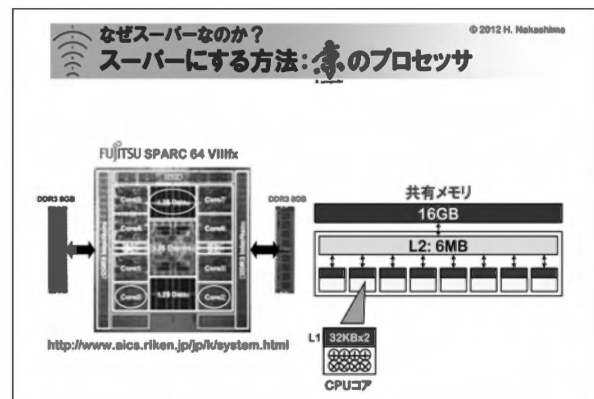
だから、同時に独立して性能の高い計算をするんですけども、全く関係ない計算をしちゃうと大体役に立ちませんので、何かの意味で話し合いをしなければいけない。そのときに、このコンピュータがここに書いて、このコンピュータが、そこから読むというと、お話ができるというわけです。これは共有メモリを使ったマルチプロセッサというように呼んでいるわけです。

こんなものを使って、計算が同時にできるんかいという話がある。じゃ、さっき3元だったんだけども、1万元ぐらいの連立方程式を解いてみようぜというわけです。そんな連立方程式あるんかいといったら、あります。1万元なんてものは、もうへっちゃらなほうで、もう最近20億円とか、何か巨大な連立方程式というのがあるわけで、解かなければいけない。そうすると、もう書くのがめんどくさいですから、何か

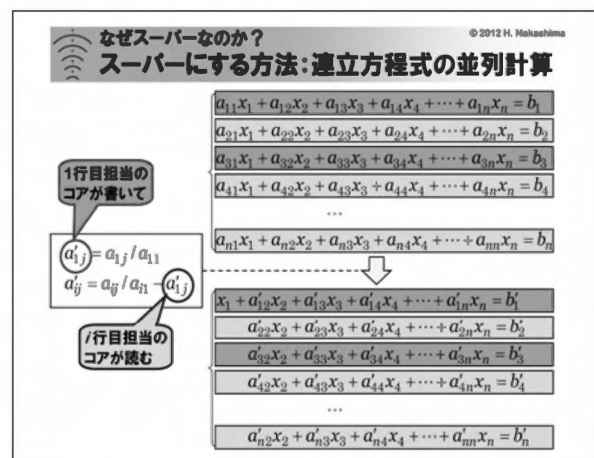
1からNみたいなことで、ウワーみたいな感じですけども、何となく分かる。(図10)

こういう連立方程式があったときに、例えば、 x を消したかったら、1行目の式を a_{11} で割ってやる。次は a_{21} で割ってやるみたいなことをやって、ほんで、こいつから、割ったやつから引いたら、何かこの辺の固まりになって、これで1つ変数が減ったわけです。 x_1 が消えた。これを繰り返してやると、どんどん、どんどん減って行って、さっきと同じように、例えば、 x_n というのが残って、それを計算して、今度逆向きにやれば良いというわけなんです。これをやっている過程として、この一番上のやつを a_{11} で割った、係数を1で割ったやつを使って、ほかの人は、それから引き算をするんですね。

つまり、自分の頭のところで一遍割っておいて、で、こいつで引いてやると、この新しいこの係数の列が見つかるということなんです。そうすると、これ1行目の担当の人が書いて、何かほかの人はそれを読むということをやれば良い。これ、たくさんありますから、



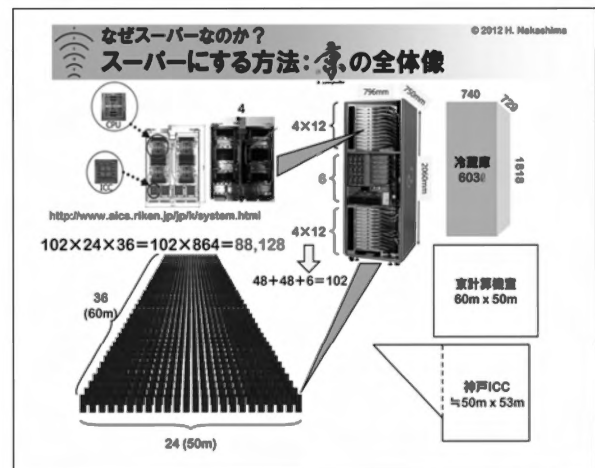
(図9)



(図10)

山のようにありますんで、何万元とかありますので、これを同時にすることができます。これで並列計算できるわけです。

じゃ、これが何個集まると京になるんだしたかという、8万8128個並べると京になります。具体的には、先ほど言ったCPUチップ、これが8個入っているやつが、何か板の上に4つ載ってまして、これを12個ここへ並べて、ここに12個並べて、ここにちょっと毛色の違うやつを6つ並べると、 $48 + 48 + 6$ 、で102個、こういうキャビネットと呼んでいますけども、タンスみたいなのが入ります。(図11)



(図11)

どのぐらい大きいかというと、ここ寸法が書いてありますが、パナソニックのwebで調べて一番でかい冷蔵庫というのが、603リットルとかいうの、そんな冷蔵庫、誰が使うんやと思うんですが、家庭用らしいんです。で、この603リットルの冷蔵庫というのが、大体同じ寸法になっています。ですから、ちょっとご家庭の冷蔵庫よりも一回り二回り大きな冷蔵庫をイメージしてもらえばいい。これを横に24、縦に36、ズワーツと864台冷蔵庫を並べると京が出来上がります。

このスペースが結構50メートル×60メートル、ちょっと縦横が逆になっていますが、この50メートル×60メートルの部屋の中にびっしり入っています。これは結構でかい部屋でして、しかも柱が1本もないという、すごい部屋なんですけど、この神戸ICCと書いてある、この建物ですが、この建物、ちょっと複雑な形状をしていて、真ん中あたりの四角い部分が50メートル×53メートルぐらいのようなので、ちょっとここよりも一回り大きいような部屋、想像がつくような、つかないような話なんですけど、だけど、そんなにびっくりするような大きな建物ではないわけです。このビルの中、巨大ではないですから、普通のビルですから。1つのビルぐらいのところの1フロアに柱が1本もないようなところをつくって、ガーッと並べると京ができます。

けど、ガーッと並べただけでは、やっぱり駄目で、つながないと駄目なんですね。つながないとお話ができない。つながり方というのが、これがまた京のホームページに行くと、6次元メッシュ/トーラス結合網、名前はTOFU(とうふ)という間抜けな名前がついているんですけども、TOFUという名前、何じゃそれはみたいな。トーラスって何や、メッシュって何や、6次元って何やねんという話なんですけど、まずメッシュというのは比較的分かりやすく、こんなふうな格好です。これ、一つ一つがコンピュータだと思ってください。この中、こういうふうな網です。網目状につながると、メッシュになります。(図12)

じゃ、トーラスって何やねんといいますが、2次元のトーラスって分かりやすいんですが、これドーナツの表面のことをトーラスと数学用語では言います。

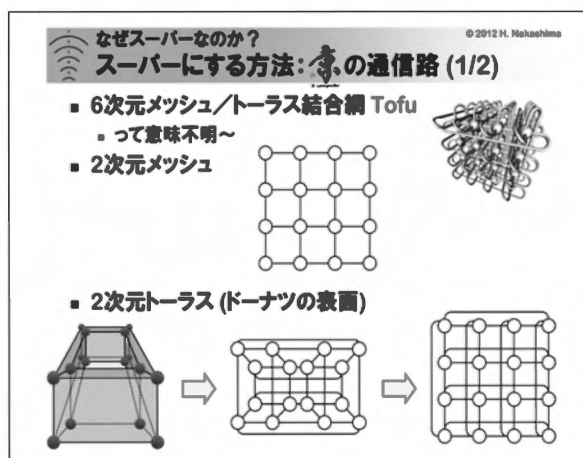
これ、ちょっとドーナツには見えませんが、とにかく中が開いた、中のほうが。この各頂点に赤い丸を配置してやって、これをぐにゅぐにゅぐにゅっと引き伸ばしてやると、だんだんこんなふうになって、こんなふうなものになります。さっきのメッシュの一番上と一番下、一番左と一番右が、またつながっているという不思議なものなんです、こういうものが2次元トーラスです。

で、6次元って何やねん。3次元は分かりますよね。3次元、我々が住んでいます。この小さい丸が京の1つのプロセッサだとすると、こっち方向に24に並んだトーラス状のグルグルと回っておる。Zのほう、高さ方向に17というちょっと変な数なんです、こうなっていて、これもグルッとこう回っている。奥行き方向には18並んでいるんですが、今度戻ってこない。行きっぱなし、だからメッシュというつながり方。これで $24 \times 17 \times 18$ ができて。(図13)

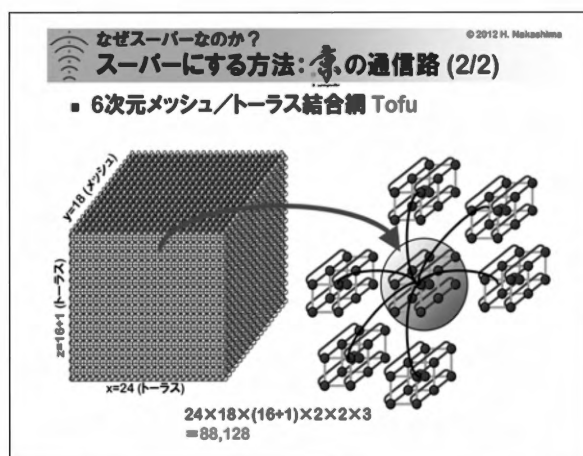
この1つは、ごめんなさい、この小さい丸はコンピュータではなくて、コンピュータが12個つながったものが、また、あるわけです。これがこうなっているというのは、この方向にトーラスになっていて、この方向にメッシュになっていると、これが3次元トーラス／メッシュと、3次元トーラス／メッシュなので、合わせて6次元トーラス／メッシュみたいな話になっているわけです。

で、何でこんなもん作ったんやという話をすると、お昼ご飯を食べる時間が全くなくなりますので、先に進みたいと思います。

さっき連立方程式をマルチコアで解いた。さっきの部屋いっぱいの京で、どうやって解くねんという、これもそう難しくなくて、さっきと同じで、これを何かこういう計算をしますと、こいつで全部パーッと1人が計算するわけなんです、それをさっきは、この結果をほかの人が読みに行けばよかったんですが、京の8万8000いくらかというのが共有



(図12)



(図13)

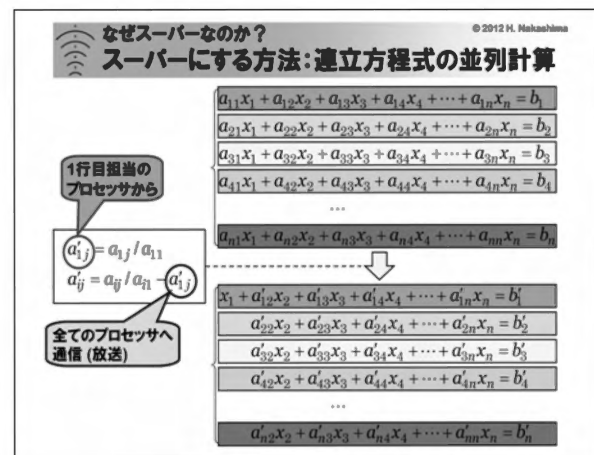
メモリになっていません。だから、電話をかけなければいけないんですね。(図14)

電話をかけるために、電話線の代わりに、さっきの6次元トラス／メッシュというのがある。どういう電話をかければいいのかというと、この場合は、この値をみんなに教えてあげればいいので、放送という電話のかけ方をします。別に電波が飛んでいるわけじゃないんですけども、もうちょっと分かりやすくいうと、分かりやすくという

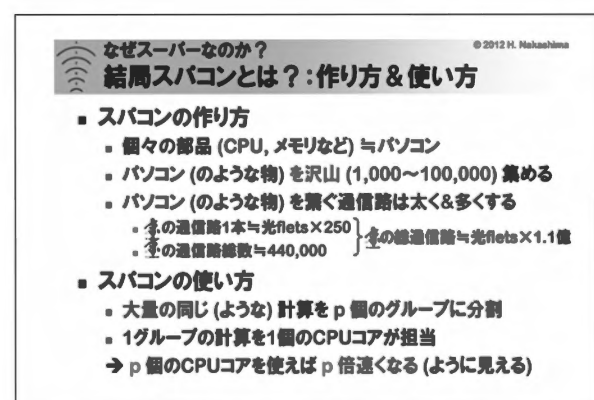
か、卑近な例でいうと、最近は駄目だそうです、クラスの連絡網というのがあって、先生から2人のお母さんに電話をかけると、そのお母さんが2人に電話をかけて、そのお母さんが、また2人に電話をかけると、40人ぐらいのクラスでは、あっという間に「あしたの遠足は中止になりました」みたいなのが行くという話があると思います。そういう感じで、この値を伝えていくような作業をします。そうすると、ここへ8万ぐらいあっても全然平気ですから、同時に計算はできるというわけです。

結局、今までのところを無理やりまとめてみますと、個々の部品、CPUチップだとか、メモリだとかいうのは、実は普通のパソコンと全然変わりません。京の場合は、秋葉原に行って、京のチップ頂戴といっても売っていませんけれども、基本的には変わらない。普通のパソコンに入っているインテルのやつを使っているスーパーコンピュータもたくさんあります。(図15)

このパソコンのようなものをたくさん、出来るだけたくさん10万個ぐらい集めましょうと。で、こいつをつなぐ、さっき6次元トラス／メッシュとかいってましたが、これを太く多くする必要があります。計算能力は一つ一つのパソコン、そこそこ計算できますけども、お話しするというのは結構大変なんですね、これが。なるべく太い線でお話ができるようにする。京の通信量1本というのが、ご家庭の光フレッツというものの250本分ぐらいの線の太さがあります。太さというんかな、要するに、昔、例えば、1メガbpsぐらいだったやつが、今は100メガbpsとか、200メガbpsになっていますけども、そういう感じでいうと、京の1本の線というのは250本分です。で、京の中には、通信の線が総数で44万ぐらいあります。



(図14)



(図15)

これとこれとを掛け算すると、光フレッツの1.1億本分。日本には1.1億世帯も絶対ありませんから、4000万世帯か3500万世帯か、そのぐらいだと思いますけど、仮に日本中にもうNTTがガッツと頑張って、あらゆる世帯に光フレッツを引いて、それでさらに日本の人口をみんなで頑張って3倍にして、世帯数を3倍にすると、大体京になるというような話なわけです。そのぐらい頑張ってつないであげると、8万何個のCPUが協力して、1つの計算をすることができる。

そういうことになっていますので、大量の出来るだけ同じような計算をp個、8万個でも100万個でもいいですけど、グループに分割します。1グループの計算を1個の、例えば、CPUコア、一番小さいコンピュータが担当してやる。そうすると、これを100万人でやると、100万倍速くなったように見える。見えるというわけです。

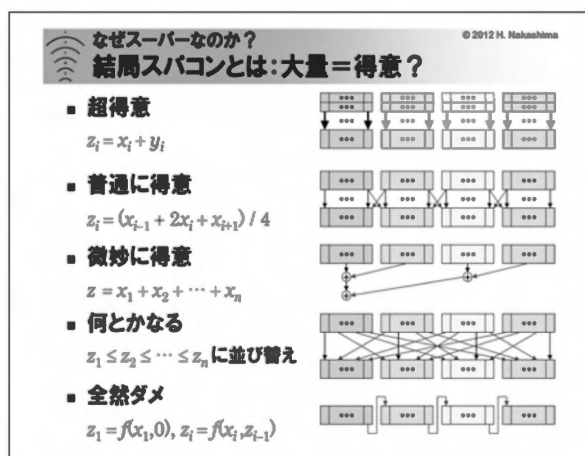
じゃ、たくさんあって分割できれば何でも計算できるか、得意か。ちょっとこれ、数式っぽく書きちゃってますけども、何かたくさん数があります。それぞれを足しましょうみたいな、これ、めちゃくちゃ得意です。これ、もうものすごく得意です。もうスーパー得意。(図16)

これちょっと変なんですけども、何かたくさん数があります。その3つをとって、真ん中2倍して、両サイドを足して4で割

る、フィルタリングみたいな、というような作業、操作になるんですが、平均化みたいなことになるんですが、これは普通です。これはどうして超得意ではないかというと、みんなで分担してやっていると、端っこのほうで、端っこ同士交換しなければいけないという話なんです。

これは分かりやすいです。たくさん数があります。みんな足しましょう。これが何かスーパー得意そうなんですけど、これが微妙に得意ぐらいなんです。なぜかという、ここ4人しかいませんが、4人で足して、4つに分けて足して、1人ずつが足した4つの答えをみんな足せばいいわけです。ほんだら足し算ができるわけなんですけど、そうすると、どうしても、この人の足し算の結果とこの人の足し算の結果を、どこかで合わさなければいけない。そうじゃないといけないですね。4つぐらいだったら平気ですが、これが100万あると、結構、向こうのほうから答えを取ってきて足さなければいけないということが、絶対どこかで生じますので、もうスーパーウルトラ得意ではない。

あと、これは数列の並び替えというようなもんです。これ、何かたくさん数があります。これ、小さい順番に並び替えてくださいと、ソーティングというんですが、これになると、結構大変なんです。たくさん数があって、並び変わった順番ってぐちゃぐちゃになってい



(図16)

ますので、ぐちゃぐちゃな何というのか、通信みたいなのが起これと、これはちょっと困る。

あとこれは、最後はちょっと数式がややこしいので、省略しますが、何か計算しましたと、その結果を使って、次の数、その結果と次の数を使って、計算しましょうと、その結果と次の数を使って計算しましょうと、この同時に計算ができない例というのが、世の中には結構あったりするんで、これはもうどうしようもない。もうこれはパソコンでやっても、京でやっても同じだけの時間がかかる。こういうようなことになっている。


問題は、こういう得意な計算というので、何かいいことができるかという話なんです、計算科学というのは、第3の科学といわれていまして、第3というのは理論科学と実験科学に続く新しい科学だといっています。(図17)

これは実施することができないとか、もう絶対できないとかいうような実験をコンピュータの中で仮想的にやる。これをシミュレーションというんですけども、これで

す。この実施困難不可能とかいっているのは、どういうことかということ、例えば、観測が絶対できないような空間というのがあります。例えば、星の内部とか、あるいは原子とか分子のレベルというのを顕微鏡で見たって、なかなか1原子は見えてこないみたいな、原子の中とかは絶対見えないみたいな話がありますので、そういうものだから、到達ができないような時間。過去、100年前のことを再現しようと思ったって、100年前に戻るわけにいかないし、さらにできそうにないのは100年後に行く、そういうような話。だから、過去の再現だとか、未来の予測というのは実験室ではできません。

それから、構築できないぐらいのでかいもの、宇宙全体とか、地球全部とか、日本全部とか、例えば、温暖化のシミュレーションとかやるんですけども、CO₂の濃度を3%下げると、どうなるでしょう。じゃ、下げて実験してみよう、地球全部でと。そんなん、できるはずがないわけです。こういうものを仮想的にコンピュータの中でやると、例えば、最初のお話でバイオリファイナーでCO₂が3%下がるとどれだけ温度が下がるとか、体重が減るとか、何とかいうのが分かる。予測することができるかもしれないというようなことです。

これらのシミュレーションというのは、実は比較的単純で基礎的な方程式の膨大な計算、ようやく大学レベルの式が出てきましたが、これを偏微分すると、何かこんなことがあるんですけども、これだけ見ると頭が痛いんですが、これを離散的に近似してやりますと、何かfという関数のtプラスデルタtというものからfの関数のtを引いてやって、それ



未来は拓くか？
計算科学＝第3の科学

■ 理論科学・実験科学に続く新しい科学
→ 実施困難・不可能な実験をコンピュータの内部で仮想的に実施 (シミュレーション)

- 観測困難・不可能な空間 (星の内部、原子・分子レベル...)
- 到達困難・不可能な時間 (過去の再現、未来の予測...)
- 構築困難・不可能な実験規模 (宇宙、地球、日本全土...)

■ (比較的単純で) 基礎的な方程式の膨大な計算

- 方程式を空間的・時間的に離散化 (近似)

(例) $\frac{\partial f(t)}{\partial t} \approx \frac{f(t+\Delta t) - f(t)}{\Delta t}$

- 刻み幅↓ ⇒ 近似精度↑
- 空間・時間規模↑ ⇒ 仮想実験の有用性↑

→ 膨大な計算 ⇒ 高精度 & 有用な仮想実験

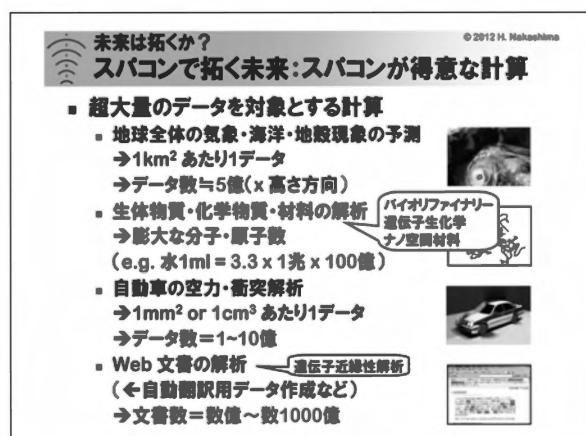
(図17)

をデルタ t で割ってやれば、大体こんなもんやねとかいうことになっていまして、大体こんなものなんです。

これをたくさんやる。刻み幅、例えば、デルタ t というのを小さくしてやると、これはほんとに連続な式なので、このデルタ t とかでは近似しかできないんですけど、近似精度がどんどん上がっていくと。あるいは空間とか、時間の規模を増やしてやると、仮想実験の有用性がどんどん上がってくる。

そうすると、要するに、刻み幅を小さくすると、空間的、時間的に離散化していますから、何かたくさんタマネギを刻むように、ガーッと刻む。それをでっかいタマネギを刻んでやると、膨大な計算というのができて、これで高精度で有用な仮想実験ができるんじゃないかということでスパコンは受けているわけです。

具体的に、もうちょっとだけ具体的に、どんなお話があるかということ、非常に大量なデータを対象とする計算です。例えば、地球全体の気象海洋地殻現象の予測というのがやりたいとして、やっていますけれども、1平方キロメートル、1データあったとすると、データ数は5億個ぐらいになります。地球全部で。1平方キロメートルでやるとするのは、そんなに精度のいい計算ではありませんので、今もうちょっと増やそうとしていますけれども、これに気象の場合、高さが加わりますので、さらにたくさんあります。(図18)



(図18)

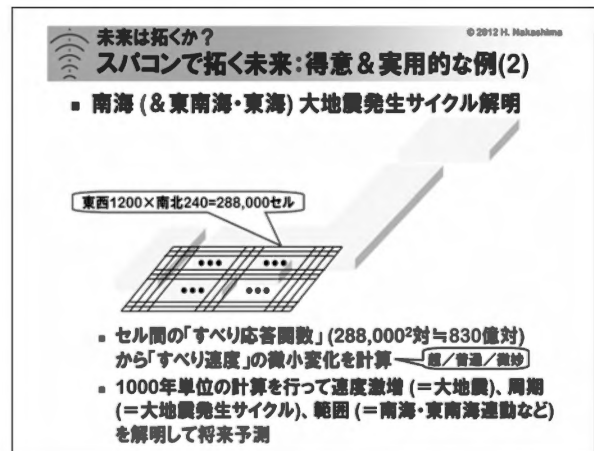
それから、例えば、生体物質、化学物質、材料の解析、きょうのバイオリファイナリーとか、遺伝子の生化学の話だとか、あるいは、午後にお話があるナノ空間材料とか、そういう話は多分この辺にかかわってくると思うんですが、例えば、水1ミリリットル取っただけで、これだけたくさんの水分子がある。これ、アボガドロ数という何か高校でやったやつを18か何かで割ると、この値になるんですけども、こういうようなものになる。

これ、実はスパコンでも、今、水1ミリリットルレベルの分子の解析なんて、全然できません。その100万分の1でも怪しいぐらいですから、非常に大変。あるいは自動車の空力とか、衝突の解析というの、トヨタさんとかやられているわけですけども、自動車を細かく切って、表面1平方ミリメートルで切ったり、あるいは中を1立方センチメートルで切ると、大体データが億のレベルになる。

あるいは、ウェブの文書ってたくさん今、もう山のようにあるんですけども、これ数億、数千億かもしれない。ちょっと数がいまいなんですが、こういうものを解析する。ここに遺伝子塩基編成解析と書いていますが、遺伝子というのは案外そういうもんでして、先

ほど何億塩基対だったかな、30億塩基対だったと思いますけども、それを文字数にすると、日本語の漢字で表現すると、4億文字ぐらいになると思うんです。その4億文字がみんな違う。ちょっとずつ違う4億文字の文章があったとして、それをたくさん比較するみたいな、こういうのは結構重要なテーマとしてあります。

もうちょっとだけ、詳しい説明をしてみます。東北で地震が起こると、次は南海やでということになっていまして、ゲーッみたいなことなんですが、南海で、それにうっとうしいことに東南海と東海が全部連動しているみたいなのがあるらしい。で、この大地震発生サイクルというのが、どうもあって、何百年に一回か、すごいのが起こるらしいんですが、これを解明するために、この辺、危なそうなところを東西1200、南北240ぐらいに分割すると、これは28万8000個の小さい四角ができます。(図19)



(図19)

この辺がズルッと滑ると、この辺でまたそれがちょっと伝わって、ズッと滑るみたいなのがあるらしくて、そのセル間の滑り応答関数みたいな、こっちが滑ると、あっちが滑る。これが、ここに28万8000あります。こっちとあっちというのが28万8000の2乗ペアあるわけです。これが830億個ぐらいありまして、これから、これをこっち側で、あっちがこうとかいうのを全部足してやったりなんかすると、各部分がズルッ、こんだけチョロッと滑るみたいなのがあると、これ、まあまあできる計算です。

で、これを1000年単位の計算をやりまして、ズルッ、チョロッと滑っていくやつが、どこかでズーッと滑るというわけです。これが南海大地震というやつで、そうすると、これが何年に一回ぐらい起こるのかとか、その周期だとか、あるいは、どこが一番壊れるんだ。要するに、南海だけなのか、南海・東南海が連動するのかというものを解明して、将来予測をしようと、できたとは言っていない。しょうです。しょうという研究が、例えば、スパコンを使って行われているというわけです。

今さっきのグラフにあったんですけど、スパコンの性能の向上率というのは、実は年間2倍弱ぐらいの率で伸びていまして、これまで過去、とんでもない率で伸びているんですが、今10ペタフロップス。これ、去年できたんですけども、そうすると、1エクサフロップスというのがペタの次に来まして、10の18乗なんですけど、これが2020年ぐらいになる。(図20、21)

ところが、これ、京を基準にしてやると、この2ギガヘルツというのは変わらない、も

上がりませんと私は言ったわけです。そうすると、この掛け算の数を増やすしかないんです。ここを8万とかいっているのを、ええい、200万にしてみまえと、この辺が8とかかわいいのが、32とか、128とかでやると速くなるんですけども、この上昇率に応じた性能アップが図れるんですが、要するに、あと6、7年で京の100倍のものをつくらんとあかんと。

さらにこれがまた続くと、ゼタフロップスというのがエクサの次に来まして、これは京の10万倍ぐらいになっていまして、ヨタフロップスって、ほんとに与太話みたいな、ほんとにあるんですけども、そういう単位が、この京のこれだけ倍になって、これがそんなに遠くないんです。2043年とか、仮に2046年だと、私はまだ90歳。まだ90歳という言い方は変ですが、ここにおられる方の半分ぐらいは生きているかもしれないというような時代に、そんなことになっちゃうかもしれない。

これが結構大変で、並列度を上げるというのは、問題がどんどん大きくなっていて、プログラムを書くのがどんどん難しくなっていくという話と、もう一つトランジスタの数が当然100倍になるんですけども、トランジスタの数が増えると、故障率が上がっちゃう。あとトランジスタの電力というのが100分の1になってくれると、トントンなんですけど、それが、なかなかそうはいかない。そうすると電力が上がると、それは困るので、故障してもいいような、あるいは、電力を抑えるようなプログラムをやろうとすると、もうどんどんまたプログラムが難しくなる。要するに、今後のスパコンというのは、プログラムをどうやってうまく書いていくかというのがカギになってくるわけです。(図22)

ここだけ私の自分の研究の話を1行だけ話させていただきますが、プログラムというのは、こういう計算をやりたいというものと、こうやって計算をやりたい。何をやりたいの



(図2.1)

未来は拓くか?
スパコン自身の未来: EFLOPSへの道

- スパコン性能向上率与×1.7~1.9/年
10PFLOPS@2011 → 1EFLOPS@2018-20
- $\Delta = 2\text{GHz} \times 8 \times 32 \times 2,000,000$ Δ の延長
 - $2\text{GHz} \times 32 \times 32 \times 500,000$ SIMD幅拡大
 - $2\text{GHz} \times 32 \times 128 \times 125,000$ コア単純化
- いずれにせよ並列度 = $\Delta \times 100$
- さらにこれが続くと
 - 1ZFLOPS = $\Delta \times 100,000$ @2031-34
 - 1YFLOPS = $\Delta \times 100,000,000$ @2043-48
 - ...

(図2.1)

未来は拓くか?
スパコン自身の未来: EFLOPSへの険路

- 並列度×100
 - 問題規模×100 → 計算量 > ×100 (×500~×10000)
 - 並列度↑ → システム複雑度↑ → プログラム複雑度↑
 - 問題規模↑ × プログラム複雑度↑ → 生産性↓↓
- トランジスタ数×100
 - トランジスタ数↑ → 故障率↑
 - 電力/トランジスタ > 1/100 → システム電力↑
 - 故障に耐える・電力を抑えるプログラム → 生産性↓↓
- 実現の鍵 = プログラム生産性↑↑
 - 「どう計算をする (what)」と「どう計算をする (how)」の分離

(図2.2)

かと、どうやってやるかという、2つの側面がありまして、難しくなるというのは、どうやって計算するかというのが、どんどん難しくなっていくというようなので、これをうまく分離してやるというのがカギだろうと、細かい話はもう避けますが、そういうことをやっています。

最後のスライドです(図23)。スパコン、25年前のスパコンというのは携帯電話よりも遅かった。20年前のスパコンはパソコンよりも遅かった。そうすると、京がパソコンとか、スマホになる日が来るかもしれない。そうすると、それは20年後から25年後の価値が新しい価値の創造なんだろうと。要するに、昔大きなコンピュータでやっていたことが、今はポケットの中

の携帯電話とか、スマホとか、パソコンとかでできるようになったので、何かいいことが起こったかという微妙ですけども、結構いろんな面白いことができるようになった。

じゃ、今から京を頑張ってパソコンにするとか、スマホにするとか、研究開発すればいいじゃんと思うかもしれませんが、それがなかなか難しくて、今まで、どうしてこんなパソコンとか、スマホが高性能化してきたかというと、10年前、20年前、30年前のスパコンから始まって、それ、どんどん、どんどん性能を上げていく努力、研究開発をしてきたわけです。その過程でパソコンというものの性能が、ある意味で上がってきた。

要するに、フロンティアの困難な課題に挑戦することで、パソコンとか、スマホとか、何万円とか、下手したら何千円とかいうようなやつのが性能が何かガーンと上がって、スマホでテレビが見られるというような時代になってきた。

ですから、例えば、こういうような大それた研究開発をすることによって、初めて何ができるか知りませんが、20年経ったときに、スマホで何ができるんでしょうね。分かりませんが、何かもっと人間の社会、あるいは生活に価値を生み出すようなことができるようになるんじゃないか、そういうふうに思っております。以上でございます。ありがとうございました。(拍手)

未来は拓く？
スパコン自身の未来: 険路を登る意義

© 2012 H. Nakashima

- コンピュータの性能↑=新たな価値創造の源泉
 - ・ スパコン@25年前<携帯電話@今
 - ・ スパコン@20年前<パソコン@今
 - 今のパソコン・スマホ化=20~25年後の価値創造
- 今のパソコン・スマホ化研究開発≠技術創出
 - ・ スパコン(性能)@1980-90のパソコン・スマホ化
 - = その後の性能↑のための研究開発の成果
 - ・ フロンティアでの困難な課題への挑戦
 - = 小型・安価なコンピュータ実現技術の源泉
 - ・ 100,000倍@20年後を目指した研究開発
 - 今のパソコン・スマホ化の実現

(図23)